

I Claim:

1. A computer apparatus suitable for use in the fast compilation of preverified platform neutral bytecode instructions resulting in high quality native machine code, comprising:

a central processing unit (CPU);

a computer memory coupled to said CPU, said computer memory comprised of a computer readable medium;

a compilation program embodied on said computer readable medium, said compilation program comprising:

a first code segment that receives a class file listing;

a second code segment that compiles said class file listing into machine code; and

a third code segment that interprets and executes said machine code.

2. A computer apparatus suitable for use in the fast compilation of preverified platform neutral bytecode instructions resulting in high quality native machine code, comprising:

a development or target computer system, said development or target computer system comprised of a computer readable storage medium containing a compilation program and one or more class files, said one or more class files containing one or more methods containing bytecode instruction listings;

said compilation program contained on said storage medium comprised of a first plurality of instructions, said first- plurality of instructions executed sequentially for each bytecode instruction, said first plurality configured to select first class to compile;

said compilation program contained on said storage medium comprised of a

second plurality of instructions, said second plurality configured to select
first method of said first class to compile;

said compilation program contained on said storage medium comprised of a third plurality of instructions, said third plurality configured to create map storage to store actual mappings and native code addresses and initialize stack mappings to empty and addresses to unknown;

said compilation program contained on said storage medium comprised of a fourth plurality of instructions, said fourth plurality configured to sequentially select each bytecode instruction in each said method of each said class file;

said compilation program contained on said storage medium comprised of a fifth plurality of instructions, said fifth plurality configured to detect stored stack maps for said selected bytecode instruction;

said compilation program contained on said storage medium comprised of a sixth plurality of instructions, said sixth plurality configured to detect direct control flow from a bytecode instruction previous to said selected bytecode instruction, said detection of direct control flow from said previous bytecode instruction resulting in said sixth plurality of instructions storing all stacks and setting said stack mappings to stack, lack of said detection of said direct control flow from said previous bytecode instruction resulting in said sixth plurality of instructions reading stack layout from said stack mappings and setting said mappings to stack;

said compilation program contained on said storage medium comprised of a seventh plurality of instructions, said seventh plurality configured to set

Attorney Docket No. 20296-300201

said native code address for actual instruction;

said compilation program contained on said storage medium comprised of an eighth plurality of instructions, said eighth plurality configured to detect if said actual instruction is a load constant instruction, said detection of load constant instruction resulting in said eighth plurality of instructions creating new constant stack mapping;

said compilation program contained on said storage medium comprised of a ninth plurality of instructions, said ninth plurality configured to detect if said actual instruction is a load local instruction, said detection of load local instruction resulting in said ninth plurality of instructions creating new local stack mapping;

said compilation program contained on said storage medium comprised of a tenth plurality of instructions, said tenth plurality configured to detect if said actual instruction is a stack manipulating instruction, said detection of stack manipulating instruction resulting in said tenth plurality of instructions duplicating or reordering said stack mapping according to said stack manipulating instruction;

said compilation program contained on said storage medium comprised of an eleventh plurality of instructions, said eleventh plurality configured to detect if said actual instruction is a jump or switch instruction, said detection of jump or switch instruction resulting in said eleventh plurality of instructions emitting code using said stack mapping information and storing all said stack values not used;

said compilation program contained on said storage medium comprised of a

twelfth plurality of instructions, said twelfth plurality configured to detect
 if said actual instruction is a remaining type of instruction, said detection
 of remaining type of instruction resulting in said twelfth plurality of
 instructions emitting code using said stack mapping information;

5 said compilation program contained on said storage medium comprised of a
 thirteenth plurality of instructions, said thirteenth plurality configured to
 select next instruction;

10 said compilation program contained on said storage medium comprised of a
 fourteenth plurality of instructions, said fourteenth plurality configured to
 select next method; and

15 said compilation program contained on said storage medium comprised of a
 fifteenth plurality of instructions, said fifteenth plurality configured to
 select next class file.

3. A computer implemented method for compilation of preverified platform neutral
 bytecode instructions resulting in high quality native machine code, comprising the steps
 of:

20 receiving a class file onto a computer readable medium containing compilation
 procedure instructions, said class file containing one or more methods
 containing platform neutral bytecode listings;
 executing said compilation procedure instructions on said bytecode listings, said
 compilation procedure instructions sequentially processing each byte code
 instruction of said bytecode listing;
 using information from preceding instructions to mimic an optimizing compiler;
 25 and

producing native machine code on said computer readable medium.

4. A computer implemented method as recited in Claim 3 wherein said compilation procedure selects first class to compile.

5

5. A computer implemented method as recited in Claim 4 wherein said compilation procedure selects first method of said first class to compile.

10

6. A computer implemented method as recited in Claim 5 wherein said compilation procedure creates map storage to store actual mappings and native code addresses and initializes stack mappings to empty and addresses to unknown.

15

7. A computer implemented method as recited in Claim 6 wherein said compilation procedure sequentially selects each bytecode instruction in each said method of each said class file.

20

8. A computer implemented method as recited in Claim 7 wherein said compilation procedure detects stack maps for said selected bytecode instruction.

25

9. A computer implemented method as recited in Claim 8 wherein said compilation procedure detects direct control flow from a bytecode instruction previous to said selected actual bytecode instruction, said detection of direct control flow from said previous bytecode instruction resulting in storing all stacks and setting said stack mappings to stack, lack of said detection of said direct control flow from said previous bytecode instruction resulting in reading stack layout from said mappings and setting said mappings to stack.

10. A computer implemented method as recited in Claim 9 wherein said compilation procedure sets said native code address for actual instruction.

11. A computer implemented method as recited in Claim 10 wherein said compilation procedure detects if said actual instruction is a load constant instruction, said detection of load constant instruction resulting in said method creating new constant stack mapping.

12. A computer implemented method as recited in Claim 11 wherein said compilation procedure detects if said actual instruction is a load local instruction, said detection of load local instruction resulting in said method creating new local stack mapping.

13. A computer implemented method as recited in Claim 12 wherein said compilation procedure detects if said actual instruction is a stack manipulating instruction, said detection of stack manipulating instruction resulting in said method duplicating or reordering said stack mapping according to said stack manipulation instruction.

14. A computer implemented method as recited in Claim 13 wherein said compilation procedure detects if said actual instruction is a jump or switch instruction, said detection of jump or switch instruction resulting in said method emitting native machine code using said stack mapping information and storing all said stack values not used.

15. A computer implemented method as recited in Claim 14 wherein said compilation procedure detects if said actual instruction is a remaining type of instruction, said detection of remaining type of instruction resulting in said method emitting native machine code using said stack mapping information.

16. A computer implemented method as recited in Claim 15 wherein said compilation procedure selects next instruction.

	1990	1991	1992	1993	1994	1995	1996	1997	1998	1999	2000	2001	2002	2003	2004	2005	2006	2007	2008	2009	2010	2011	2012	2013	2014	2015	2016	2017	2018	2019	2020	2021	2022	2023	2024	2025	2026	2027	2028	2029	2030	2031	2032	2033	2034	2035	2036	2037	2038	2039	2040	2041	2042	2043	2044	2045	2046	2047	2048	2049	2050	2051	2052	2053	2054	2055	2056	2057	2058	2059	2060	2061	2062	2063	2064	2065	2066	2067	2068	2069	2070	2071	2072	2073	2074	2075	2076	2077	2078	2079	2080	2081	2082	2083	2084	2085	2086	2087	2088	2089	2090	2091	2092	2093	2094	2095	2096	2097	2098	2099	2100	2101	2102	2103	2104	2105	2106	2107	2108	2109	2110	2111	2112	2113	2114	2115	2116	2117	2118	2119	2120	2121	2122	2123	2124	2125	2126	2127	2128	2129	2130	2131	2132	2133	2134	2135	2136	2137	2138	2139	2140	2141	2142	2143	2144	2145	2146	2147	2148	2149	2150	2151	2152	2153	2154	2155	2156	2157	2158	2159	2160	2161	2162	2163	2164	2165	2166	2167	2168	2169	2170	2171	2172	2173	2174	2175	2176	2177	2178	2179	2180	2181	2182	2183	2184	2185	2186	2187	2188	2189	2190	2191	2192	2193	2194	2195	2196	2197	2198	2199	2200	2201	2202	2203	2204	2205	2206	2207	2208	2209	2210	2211	2212	2213	2214	2215	2216	2217	2218	2219	2220	2221	2222	2223	2224	2225	2226	2227	2228	2229	2230	2231	2232	2233	2234	2235	2236	2237	2238	2239	2240	2241	2242	2243	2244	2245	2246	2247	2248	2249	2250	2251	2252	2253	2254	2255	2256	2257	2258	2259	2260	2261	2262	2263	2264	2265	2266	2267	2268	2269	2270	2271	2272	2273	2274	2275	2276	2277	2278	2279	2280	2281	2282	2283	2284	2285	2286	2287	2288	2289	2290	2291	2292	2293	2294	2295	2296	2297	2298	2299	2300	2301	2302	2303	2304	2305	2306	2307	2308	2309	2310	2311	2312	2313	2314	2315	2316	2317	2318	2319	2320	2321	2322	2323	2324	2325	2326	2327	2328	2329	2330	2331	2332	2333	2334	2335	2336	2337	2338	2339	2340	2341	2342	2343	2344	2345	2346	2347	2348	2349	2350	2351	2352	2353	2354	2355	2356	2357	2358	2359	2360	2361	2362	2363	2364	2365	2366	2367	2368	2369	2370	2371	2372	2373	2374	2375	2376	2377	2378	2379	2380	2381	2382	2383	2384	2385	2386	2387	2388	2389	2390	2391	2392	2393	2394	2395	2396	2397
--	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------

10

10